

AMENDMENTS TO THE CLAIMS

For the convenience of the Examiner, all claims have been presented whether or not an amendment has been made. The claims have been amended as follows:

1. **(Currently Amended)** A method for interfacing a **computerized** genetic search algorithm to the web environment of the Internet, comprising:

defining data elements, attributes and rules for use thereof for an extensible markup language;

storing the extensible markup language in a document type definition file **on a computer**;

describing data in a document, **on the computer**, in a hierarchical format utilizing the stored extensible markup language; and

importing the data in the document to a genetic algorithm **comprising a randomized search algorithm** to define a data string or an individual in a population of points, **the population of points comprising a bit string encoded to represent the data or individual**.

2. **(Original)** A method for interfacing a genetic search algorithm as in Claim 1 wherein defining data elements, attributes and rules comprises defining a tree-based crossover operator.

3. **(Original)** A method for interfacing a genetic search algorithm as in Claim 1 wherein defining data elements, attributes and rules comprises defining a mutation operator.

4. **(Original)** A method for interfacing a genetic search algorithm as in Claim 3 further comprising importing elements from the document definition file as nodes of the stored extensible markup language.

5. **(Original)** A method for interfacing a genetic search algorithm as in Claim 4 wherein defining a mutation operator comprises performing the mutation operator either at an internal or external node.

6. **(Original)** A method for interfacing a genetic search algorithm as in Claim 1 wherein defining data elements, attributes and rules comprises defining a permutation operator, an editing operator, an encapsulation operator, and a dissemination operator.

7. **(Currently Amended)** A method for interfacing an extensible markup language computerized genetic search algorithm to the web environment of the Internet, comprising:

providing a genetic algorithm parameter document, on a computer, containing input data;

creating a population of individuals from the input data;

randomly generating documents, on the computer, conforming to the genetic algorithm individual DTD; and

providing utilities, on the computer, for performing crossover operations and/or mutation operations.

8. **(Original)** A method for interfacing a genetic search algorithm as in Claim 7 wherein providing a parameter document comprises providing constraints for the created individuals.

9. **(Original)** A method for interfacing a genetic search algorithm as in Claim 8 wherein providing constraints for the created individuals comprises providing one or more constraints from the group comprising: uniqueness, completeness and maximum and minimum values for nodes that contain text data.

10. **(Original)** A method for interfacing a genetic search algorithm as in Claim 7 wherein creating a population of individuals comprises creating programs organized into a tree structure.

11. **(Original)** A method for interfacing a genetic search algorithm as in Claim 10 wherein providing utilities comprises providing a crossover operator for structuring nodes of the tree structure.

12. **(Original)** A method for interfacing a genetic search algorithm as in Claim 10 wherein providing utilities comprises providing a mutation operator for performing on either internal or external nodes of the tree structure.

13. **(Original)** A method for interfacing a genetic search algorithm as in Claim 7 wherein providing utilities comprises providing a bit-wise crossover operator and/or a tree-node crossover operator.

14. **(Currently Amended)** A method for interfacing an extensible markup language computerized genetic search algorithm to the web environment of the Internet, comprising:

providing a genetic algorithm parameter document, on a computer, containing input data;

creating a population of individuals from the input data;

randomly generating documents, on the computer, conforming to the genetic algorithm individual DTD; and

providing utilities, on the computer, for performing one or more of the operators from the group comprising: crossover, mutation, permutation, editing, encapsulation, and dissemination.

15. **(Original)** A method for interfacing a genetic search algorithm as in Claim 14 wherein providing utilities comprises providing a permutation operator for reordering the characters found between two selected points of a single individual.

16. **(Original)** A method for interfacing a genetic search algorithm as in Claim 14 wherein providing utilities comprises providing an editing operator for recursively applying a pre-established set of editing rules to each individual member of the population.

17. **(Original)** A method for interfacing a genetic search algorithm as in Claim 14 wherein creating a population of individuals comprises creating programs organized into a tree structure.

18. **(Original)** A method for interfacing a genetic search algorithm as in Claim 17 wherein providing utilities comprises providing an encapsulation operator for identifying and naming potentially useful sub trees to enable referencing for later use.

19. **(Original)** A method for interfacing a genetic search algorithm as in Claim 14 wherein providing utilities comprising providing a dissemination operator for ridding the population of individuals of low-fitness individuals in cases where the population of individuals becomes skewed and has a high proportion of such individuals.

20. **(Currently Amended)** A method for interfacing an extensible markup language computerized genetic search algorithm to the web environment of the Internet, comprising:

providing a genetic algorithm parameter document, on a computer, containing input data;

providing an XSL style sheet containing instruction for the genetic algorithm to construct a population of individuals;

creating a population of individuals from the input data in accordance with the instructions from the XSL style sheet;

randomly generating documents, on the computer, conforming to the genetic algorithm individual DTD;

providing a fitness function for the individuals of the created population; and

providing utilities, on the computer, for performing one or more of the operators selected from the group comprising: crossover, mutation, permutation, editing, encapsulation, and dissemination.

21. **(Original)** A method for interfacing a genetic search algorithm as in Claim 20 wherein providing a parameter document comprises providing constraints for the created individuals.

22. **(Original)** A method for interfacing a genetic search algorithm as in Claim 21 wherein providing constraints comprises providing one or more constraints from the group comprising: uniqueness, completeness, and maximum and minimum values for nodes that contain text data.

23. **(Original)** A method for interfacing a genetic search algorithm as in Claim 20 wherein creating a population of individuals comprises creating programs organized into a tree structure.

24. **(Original)** A method for interfacing a genetic search algorithm as in Claim 23 wherein providing utilities comprises providing a bit-wise crossover operator and/or a tree-node crossover operator.